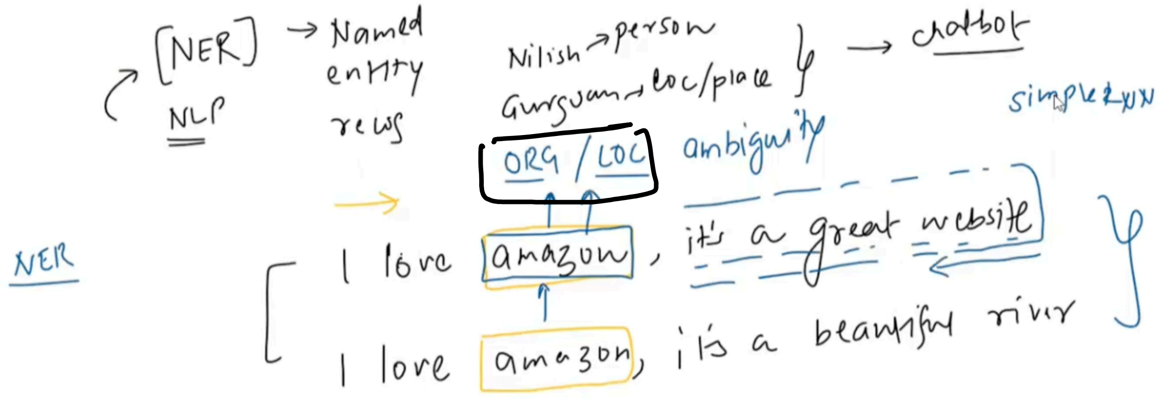


[NER] → Named entity rews

NLP

Nilish → person

Gurguam → loc/place

→ chatbot

simplezyx

ORG / LOC ambiguity

NER

I love amazon, it's a great website

I love amazon, it's a beautiful river

Bidirectional Architechure

Amazon the best website

Amazon the beautiful river.

forward

→ ←

2 RNN RNN RNN.

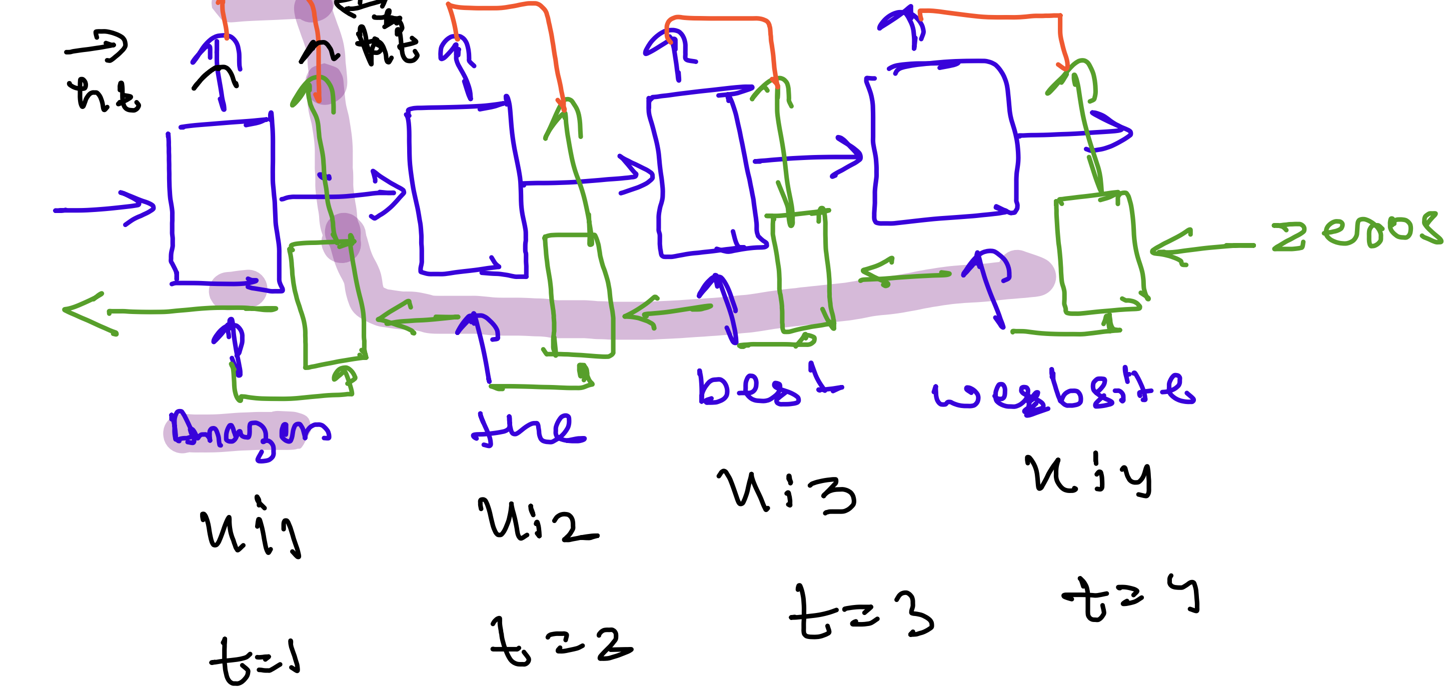
→ ←

Read left to right

Read right to left

concatinate.

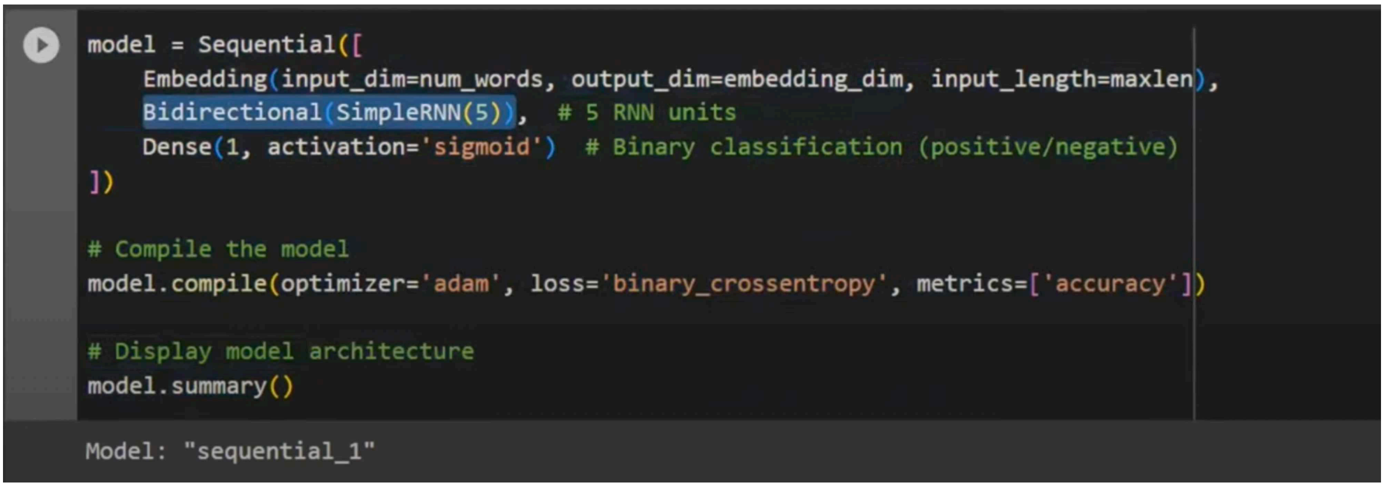
Y_1 Y_2 Y_3 Y_4



$$\overrightarrow{h_t} = \tanh(\overrightarrow{W} \overrightarrow{h_{t-1}} + \overrightarrow{U} x_t + \overrightarrow{b})$$

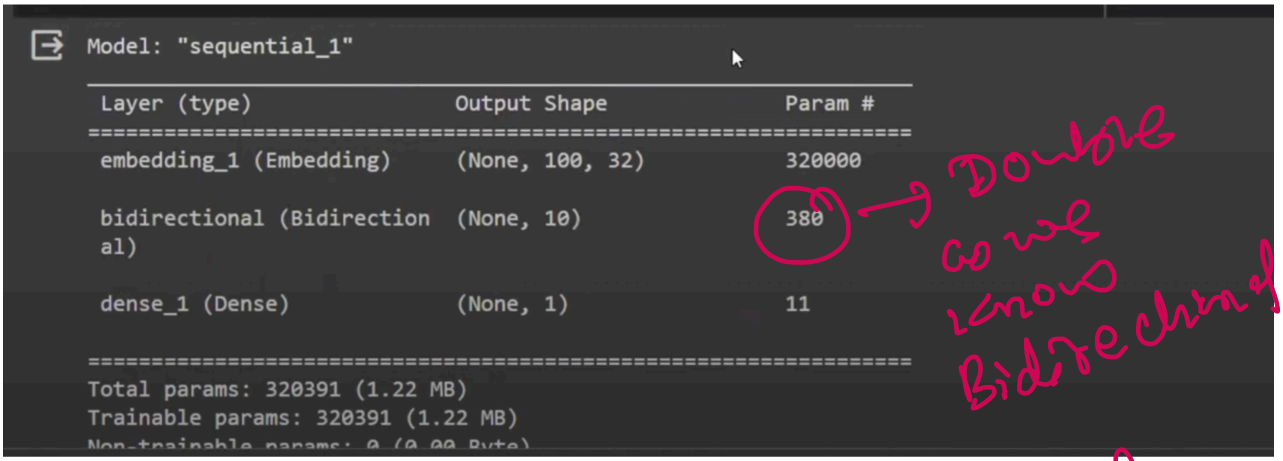
$$\overleftarrow{h_t} = \tanh(\overleftarrow{W} \overleftarrow{h_{t+1}} + \overleftarrow{U} x_t + \overleftarrow{b})$$

$$\boxed{y_t = \sigma(V[\overrightarrow{h_t}, \overleftarrow{h_t}] + b)}$$



```
model = Sequential([
    Embedding(input_dim=num_words, output_dim=embedding_dim, input_length=maxlen),
    Bidirectional(SimpleRNN(5)),  # 5 RNN units
    Dense(1, activation='sigmoid')  # Binary classification (positive/negative)
])

# Compile the model
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])

# Display model architecture
model.summary()
```

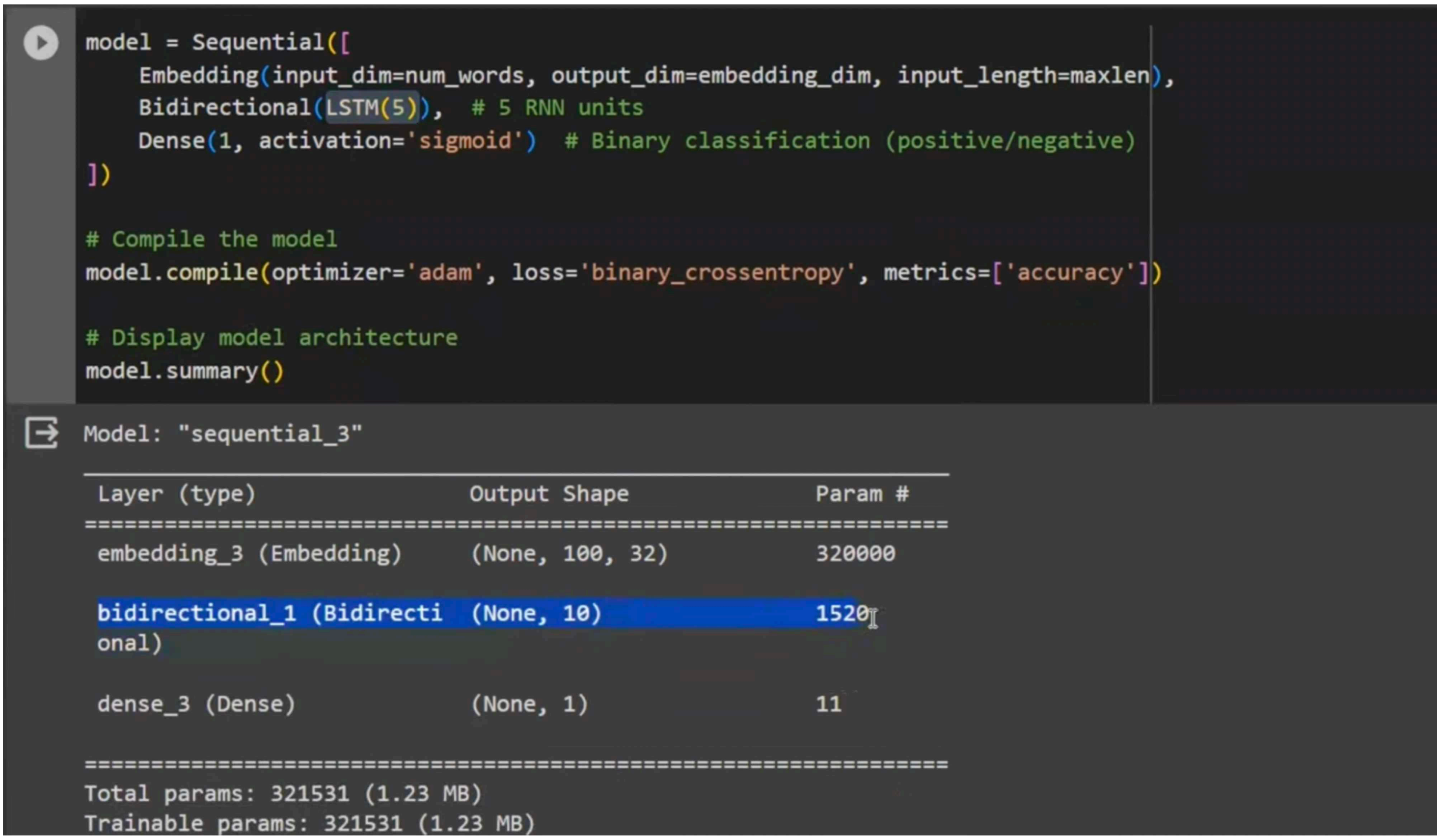
```
Model: "sequential_1"
```



```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_1 (Embedding)     (None, 100, 32)           320000

 bidirectional (Bidirection  (None, 10)                380
 al)

 dense_1 (Dense)             (None, 1)                 11

=================================================================
Total params: 320391 (1.22 MB)
Trainable params: 320391 (1.22 MB)
Non-trainable params: 0 (0.00 Byte)
```

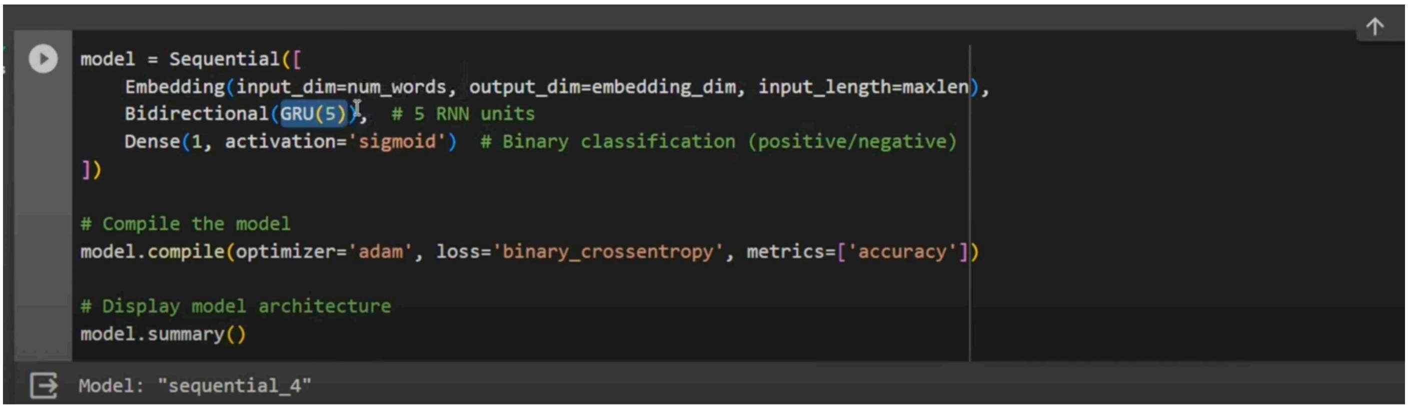
→ Double coz we know Bidirectional

RNN has 2 RNN.



```
model = Sequential([
    Embedding(input_dim=num_words, output_dim=embedding_dim, input_length=maxlen),
    Bidirectional(LSTM(5)),  # 5 RNN units
    Dense(1, activation='sigmoid')  # Binary classification (positive/negative)
])

# Compile the model
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])

# Display model architecture
model.summary()
```

```
Model: "sequential_3"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_3 (Embedding)     (None, 100, 32)           320000

 bidirectional_1 (Bidirecti  (None, 10)                1520
 onal)

 dense_3 (Dense)             (None, 1)                 11

=================================================================
Total params: 321531 (1.23 MB)
Trainable params: 321531 (1.23 MB)
```



```
model = Sequential([
    Embedding(input_dim=num_words, output_dim=embedding_dim, input_length=maxlen),
    Bidirectional(GRU(5)),  # 5 RNN units
    Dense(1, activation='sigmoid')  # Binary classification (positive/negative)
])

# Compile the model
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])

# Display model architecture
model.summary()
```

```
Model: "sequential_4"
```

Applications

1. NER
2. Part of speech tagging.
3. Machine translation.
4. Sentiment Analysis.
5. Time series forecasting

Drawbacks

1. Complexity. speech recognition
2. Need full data,

→ slow